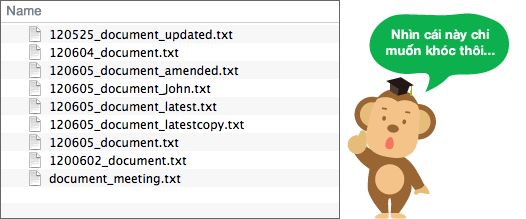
Hướng dẫn về GIT cho người mới bắt đầu

# Git cơ bản

## Mở đầu

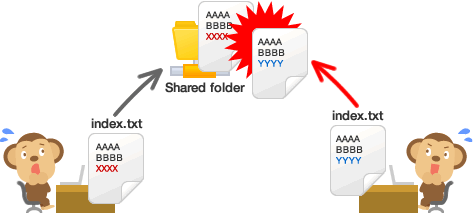
Khi muốn đưa về trạng thái trước khi chỉnh sửa của file thì mọi người sẽ làm như thế nào?

Cách đơn giản nhất đó là sao chép lại file trước khi chỉnh sửa. Trường hợp dùng phương pháp này thì sẽ phải thường xuyên thực hiện việc thêm ngày đã thay đổi vào tên thư mục hay file. Tuy nhiên, việc tự mình sao chép file mỗi lần chỉnh sửa thì sẽ rất vất vả, và cũng dễ xảy ra nhầm lẫn.



Hơn nữa, trường hợp đã đặt tên một cách vô trật tự như hình trên thì sẽ không thể phân biệt được file nào là bản mới nhất. Thêm nữa, với file đang chia sẻ và làm việc trong nhóm thì cũng sẽ có gắn tên của người chỉnh sửa. Nhưng lại không thể dễ dàng biết được đã thực hiện thay đổi như thế nào.

Ngoài ra, trường hợp file đang chia sẻ trong nhóm, nếu có hai người chỉnh sửa cùng lúc, thì sẽ xảy ra trường hợp xóa mất nội dung thay đổi của người đã chỉnh sửa trước đó.



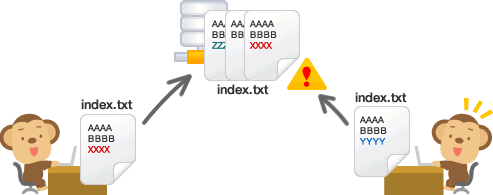
Và để giải quyết những vấn đề này thì các hệ thống quản lý phiên bản như Git được tạo ra.

## Quản lý phiên bản sử dụng Git

Git là một trong những Hệ thống Quản lý Phiên bản Phân tán, vốn được phát triển nhằm quản lý mã nguồn (source code) hữu hiệu của Linux.

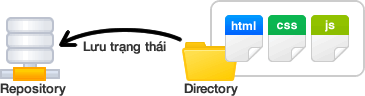
Trên Git, có thể lưu trạng thái của file khi có nhu cầu dưới dạng lịch sử cập nhật. Vì thế, có thể đưa file đã chỉnh sửa một lần về trạng thái cũ hay có thể hiển thị sự khác biệt ở nơi chỉnh sửa.

Thêm nữa, khi định ghi đè (overwrite) lên file mới nhất đã chỉnh sửa của người khác bằng file đã chỉnh sửa dựa trên file cũ, thì khi đăng (upload) lên server sẽ hiện ra cảnh cáo. Vì thế, sẽ không xảy ra thất bại về việc đã ghi đè lên nội dung chỉnh sửa của người khác mà không hề hay biết.



## Repository quản lý lịch sử

Repository là nơi sẽ ghi lại trạng thái của thư mục và file. Trạng thái được lưu lại đang được chứa như là lịch sử thay đổi của nội dung. Bằng việc đặt thư mục muốn quản lý lịch sử thay đổi dưới sự quản lý của repository, có thể ghi chép lại lịch sử thay đổi của thư mục và file trong thư mục đó.

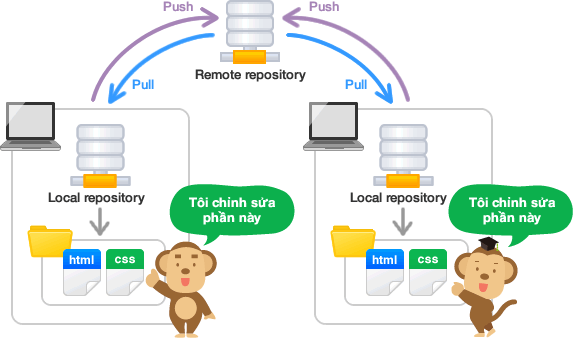


## Remote repository và local repository

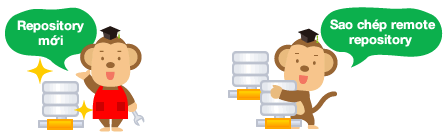
Đầu tiên, repository của Git được phân thành 2 loại là remote repository và local repository.

* ***Remote repository***: Là repository để chia sẻ giữa nhiều người và bố trí trên server chuyên dụng.
* ***Local repository***: Là repository bố trí trên máy của bản thân mình, dành cho một người dùng sử dụng.

Do repository phân thành 2 loại là local và remote nên với những công việc bình thường thì có thể sử dụng local repository và thực hiện trên toàn bộ máy sẵn có. Khi muốn công khai nội dung công việc mà bản thân đã làm trên local repository, thì sẽ upload lên remote repository rồi công khai. Thêm nữa, thông qua remote repository cũng có thể lấy về nội dung công việc của người khác.



## Tạo repository

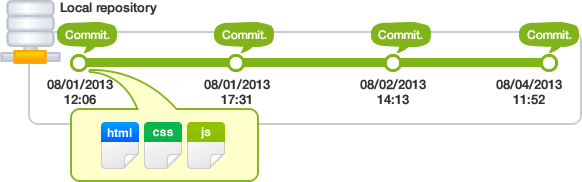


## Commit sẽ ghi lại thay đổi

Để ghi lại việc thêm/ thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit.

Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước đến trạng thái hiện tại.

Commit này đang được chứa tại repository trong trạng thái đã nối tiếp với nhau theo thứ tự thời gian như trong hình dưới đây. Bằng việc lần theo commit này từ trạng thái mới nhất thì có thể hiểu được lịch sử thay đổi trong quá khứ hoặc nội dung thay đổi đó.



Các commit này, được đặt tên bởi 40 ký tự alphabet không trùng nhau đã được tính toán từ thông tin commit. Bằng việc chỉ định tên này, có thể chỉ định commit từ trong repository.

Tip: Những thay đổi mang ý nghĩa khác nhau chẳng hạn như thêm chức năng hay sửa lỗi thì hãy cố gắng chia ra rồi commit. Để sau này khi xem lịch sử và tìm kiếm một nội dung thay đổi định sẵn sẽ dễ dàng hơn.

Khi thực hiện commit có yêu cầu nhập giải thích commit (commit message). Vì commit message là bắt buộc nên nếu để trống mà thực hiện thì commit sẽ thất bại.

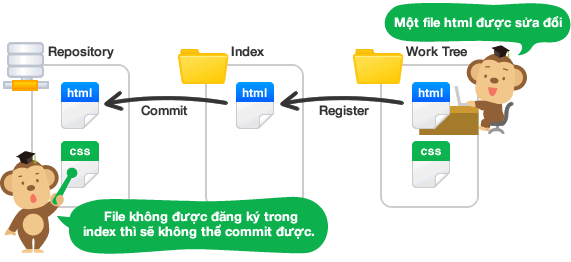
Vì phần giải thích đó sẽ trở thành thông tin quan trọng khi bản thân xem lại lịch sử sau đó hay trường hợp tìm kiếm nội dung thay đổi mà người khác commit, nên hãy chú ý để viết giải thích sao cho dễ dàng hiểu được nội dung thay đổi. Viết giải thích theo hình thức tiêu chuẩn trên Git như sau:

* Dòng thứ 1: Tóm tắt nội dung thay đổi trong commit
* Dòng thứ 2: Dòng trống
* Dòng thứ 3 trở đi: Lý do đã thay đổi

## Working Tree và Index

Trên Git, những thư mục được đặt trong sự quản lý của Git mà mọi người đang thực hiện công việc trong thực tế được gọi là working tree.

Và trên Git, giữa repository và working tree tồn tại một nơi gọi là index. Index là nơi để chuẩn bị cho việc commit lên repository.



Trên Git, khi đã thực hiện commit thì trạng thái sẽ không được ghi trực tiếp trong repository từ working tree, mà sẽ ghi trạng thái đã được thiết lập của index được xây dựng ở giữa đó. Vì thế, để ghi lại trạng thái của file bằng commit thì trước hết cần đăng ký file trong index.

Với việc chèn index vào giữa như thế này, có thể thực hiện commit không bao gồm những file không cần thiết trong working tree, hay có thể đăng ký chỉ một phần thay đổi của file trong index rồi commit.

# Hướng dẫn 1: Cơ bản của Git

## Cài đặt Git

Trước khi bước vào hướng dẫn, đầu tiên chúng ta hãy xây dựng môi trường để sử dụng Git. Ở đây sẽ chỉ định và cài đặt ứng với các môi trường Windows(GUI), Mac(GUI), Console(command line). Và trở về sau cũng sẽ sử dụng Git client đã chọn mà tiếp tục giải thích.

Tip: Với những lập trình viên hay những người đã quen thuộc với console thì nếu dùng console mà thử trải nghiệm Git sẽ tốt hơn.

### Windows

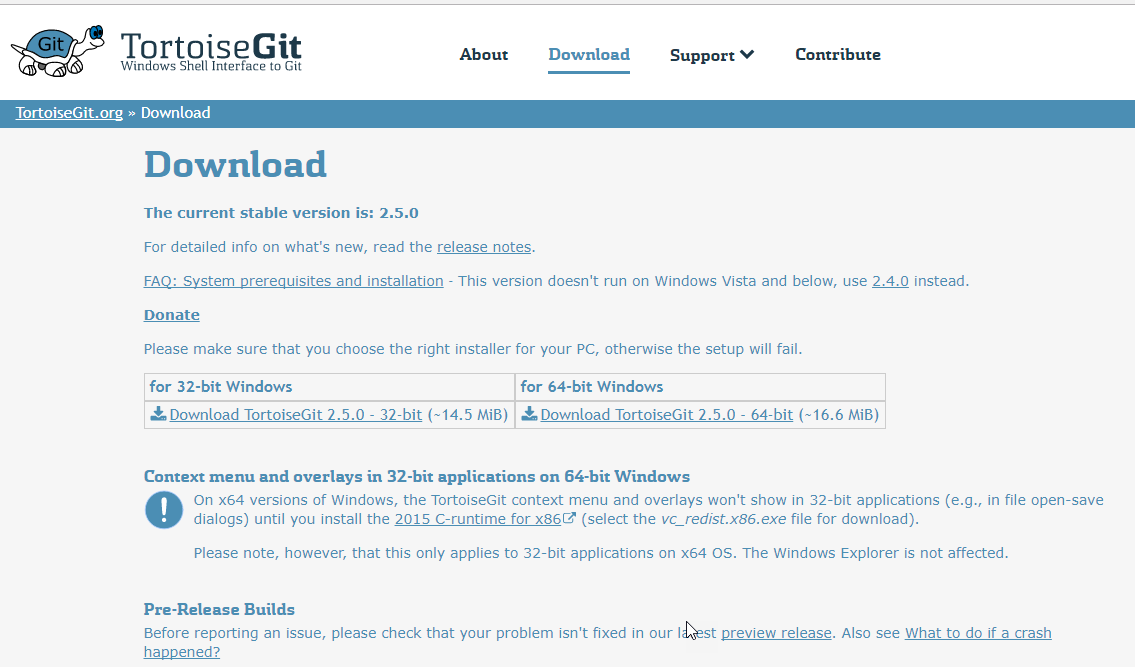
Ở đây sẽ giới thiệu về TortoiseGit, Git client của mã nguồn mở (open source). Thêm nữa, cũng sẽ giới thiệu về gói ngôn ngữ (Language Pack) sẽ hiển thị tiếng Nhật chẳng hạn như trên menu .v.v...

<http://code.google.com/p/tortoisegit/>

Lưu ý: Để sử dụng TortoiseGit, cần cài đặt msysgit trước. Từ trang web dưới đây hãy tải trình cài đặt (installer) rồi cài đặt.

<http://msysgit.github.io/>

Trước tiên, từ trang web sau đây hãy tải trình cài đặt và gói ngôn ngữ của TortoiseGit. Vì có thể tải về bản 32 bit và 64 bit nên hãy chọn bản giống với Windows mà mình đang sử dụng.

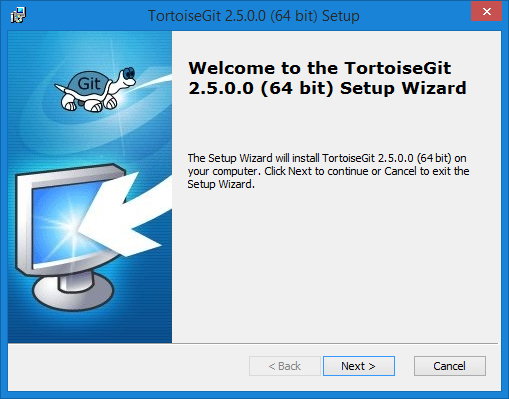


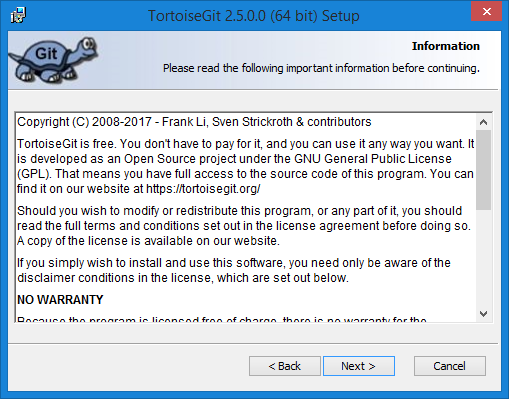
Trường hợp không biết Windows đang sử dụng là bản 32 bit hay bản 64 bit thì có thể tìm hiểu bằng cách sau đây.

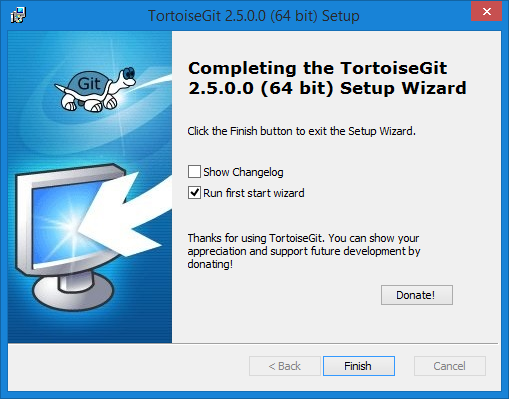
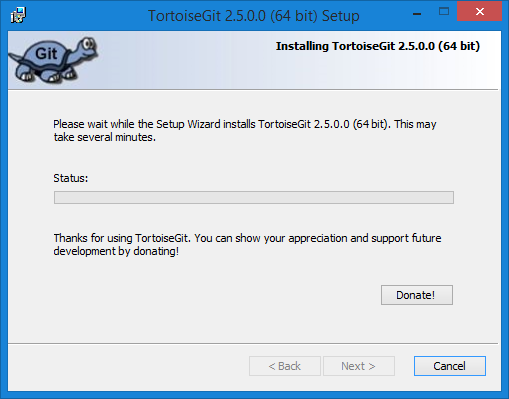
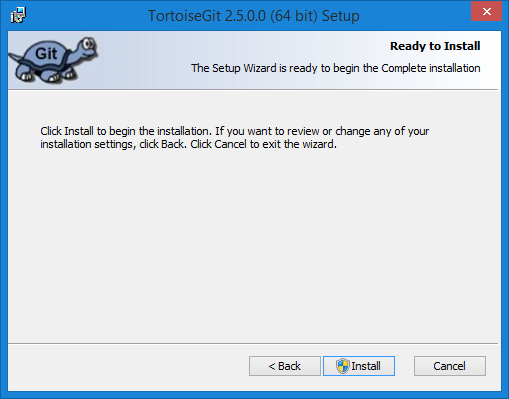
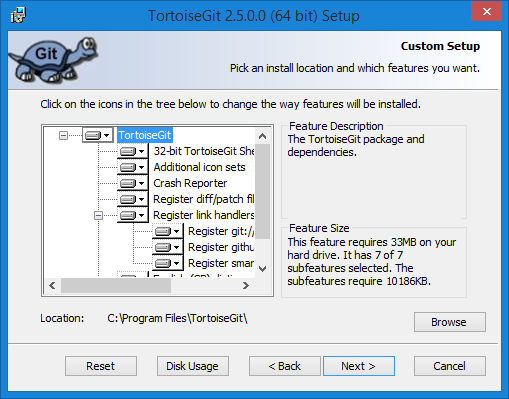
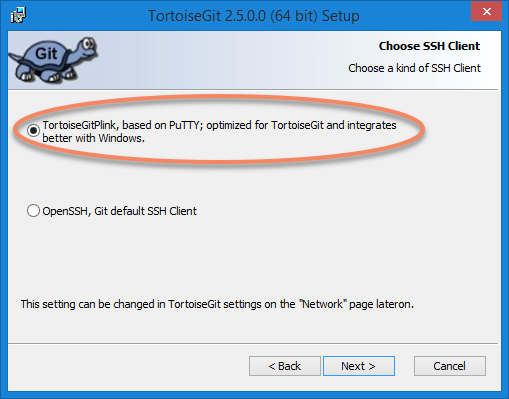
Click chuột phải vào [Computer] ở [Start menu], rồi từ menu hiện ra click vào [Properties].

Và thông tin cần biết sẽ được hiển thị ở phần [System type] trên màn hình tiếp theo.

Double click vào file trình cài đặt đã tải về rồi khởi động. Màn hình tiếp theo sẽ được hiển thị, click vào nút "Next".







Chỉnh sửa 1 xíu